Goldvin Sugirtha DHAS. B and Deepa. S.N/ IU-JEEE Vol. 16(1), (2016), 1995-2002

EVOLUTIONARY ALGORITHMS FOR PID CONTROLLER DESIGN OF BOOST INVERTER IN PHOTOVOLTAIC APPLICATIONS

Goldvin Sugirtha DHAS. B[1], Deepa. S.N[2]

[1,2]Department of Electrical and Electronics Engineering, Anna University Regional Campus, Coimbatore, India.
bgsdhas@hotmail.com, deepapsg@gmail.com

Abstract: *This paper proposes a design method for determining the optimal proportional-integral-derivative (PID) controller parameters of a boost inverter using the evolutionary algorithms for photovoltaic (PV) system. This paper details how to employ the evolutionary algorithm to search efficiently the optimal PID controller parameters of boost inverter. A time-domain performance criterion function was used in order to assist estimating the optimal PID controller parameters. Comparison is made among the genetic algorithm (GA), Ant Colony Optimization (ACO), Artificial Bee Colony algorithm (ABC) and Particle Swarm Optimization (PSO). It is found that Particle Swarm Optimization algorithm based PID controller was indeed more efficient and robust in improving the step response of a boost inverter. Simulations are carried out and results are presented to validate the performance of the PID controller.*
Keywords: *Ant Colony Optimization (ACO), Artificial Bee Colony Algorithm (ABC), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Proportional Integral Derivative (PID).*

1. Introduction

Due to the realization of problems in waste management and decommissioning cost of the nuclear power plant, the concentration towards the renewable energy sources has increased drastically. Among them, solar power generation attracts more attention because, it is pollution and radiations free and they provide excellent opportunity to generate electricity [1]. However, solar energy conversion systems generally suffer from power quality problems like harmonics and power factor due to the use of power semiconductor devices like IGBT and MOSFET, etc.

Usually in a power conversion system, power conversion takes place in two stages: the boosting stage and the inverting stage. Numerous topologies like Boost, Buck-Boost [2], Cuk [3], SEPIC [4], ZVS [5] and Resonant converter [6], etc. are used for boosting stage and multilevel and multi-pulse inverters are used for inverting stage [7]. In the two stage system, the numbers of switching devices, inductive and capacitive components are numerous leading to switching losses, which in turn reduces the efficiency of the converter [8]. There are also single stage converters available for power conversion which have a dc link capacitor and an inverter which converts dc supply directly to ac [9], [10], [11] and produces an output voltage less than the input dc.

So in order to overcome these issues, a boost inverter capable of both boosting and inverting the power in a single stage is used. The boost inverter produces an output ac voltage which can be higher or lower than the given dc input depending on the duty cycle. The number of switches required is less (four) and also the quality of the output voltage is sinusoidal. The controlling of the ac output voltage requires controlling of both boost converters. However, the boost inverter is a difficult system to be controlled. There have been few controllers reported in the literature.

In [12], the sliding mode control is proposed to control the boost inverter which can deal with variable operation point conditions and achieves good steady state results. However, it requires complex control theory, variable switching frequency, lacks inductance averaged-current control and constraints to controller parameter selection.

In [13], each boost converter is controlled by means of a double-loop control scheme that consists of an inner inductor current control loop and an outer voltage control loop. Both control loops are developed based on the averaged continuous-time model of the boost topology. The control loops include several compensations in order to decouple the converter model from the operation point of the controller. Additionally feed-forward regulators are used to keep the inverter in stable operating condition.

In [14], a single controller is proposed to focus on generating a sinusoidal voltage on the load despite the voltage in both capacitors. The resulting controller is very easy to implement without the use of current sensors. However, the controller is nonlinear due to the use of small power switches and the feedback of the control output within the control circuit itself.

In [15], for small distributed grid-connected inverters, output current control is proposed to control the exporting power independent of the load at the point of connection. it

Received on: 10.03.2015
Accepted on: 30.11.2015

requires the design of proportional-and-resonant controllers. This implementation of this control is complex as it requires multipliers, divisors, and other complicated circuitry.

In connection with these problems, the traditional PID controllers can handle boost inverter control with the help of active and passive compounds. In the case of digital control, it requires high speed processors. For the case of commercialization of controllers, the PID controllers are cost effective for controlling the boost inverter. Therefore PID controller is presented as a simple alternative to the control the switched power converters. And the PID controller is tuned using Evolutionary Algorithms (EA). The modelling of PV array is presented in Section 2. The linear model of a boost inverter system is presented in Section 3. The PID tuning using proposed optimization techniques is described in Section 4. The simulation results are presented to validate the proposed method. And the paper concludes finally.

2. Modelling of the PV Module and Array

A PV module can be represented by an electrical equivalent one-diode model as shown in Figure 1. This model contains a current source I_g, a diode D, and a series resistance R_S , which represents the resistance inside each cell and in the connection between the cells.

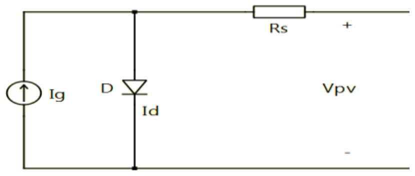


Figure 1. Single-Diode model of the PV cell

2.1. Modeling PV Module

The PV module can be modelled using the equivalent circuit as net current, I_{PV} is the difference between the photocurrent I_g and the diode current I_D and is given by,

$$I_{PV} = I_g - I_S\left(\exp\left(\frac{q(V_{PV} + I_{PV}.R_S)}{nKT}\right) - 1\right), \qquad (1)$$

where, n is the diode ideality factor, k is Boltzmann's constant ($1.3806503 \times 10^{-23}$ J/K), q is the electron charge ($1.60217646 \times 10^{-19}$ C), T is the temperature in Kelvin, R_S is the equivalent resistance and I_S is the saturation current.

2.2. Modeling PV Array

For a large power system, the PV modules are configured in series-parallel connection (i.e. $N_S \times N_P$ modules). In this case, the array is configured as 4×2 (series–parallel) resulting in the total power of 480W at STC with each solar array panel module (MSX-60) rated at 60W. The PV output current Eqn (1) is modified as,

$$I_{PV} = N_P\left\{I_g - I_S\left(\exp\left(\frac{q(V_{PV} + I_{PV}.R_S M)}{nKT}\right) - 1\right)\right\} \qquad (2)$$

and

$$M = \frac{N_S}{N_P}, \qquad (3)$$

where, N_s and N_p are the number cells connected in series and parallel respectively.

3. Linear Model of a Boost Inverter

The boost inverter consists of two boost converter with common dc photovoltaic supply and is modulated at 180◦ out of phase with each other to produces a sinusoidal voltage at the output. The load is connected differentially across the converters [6] and is as shown in Figure 2.

The voltage V_1 is across the capacitor C_1 and the voltage V_2 is across the capacitor C_2 and it is 180◦ out of phase with V_1. The load R_0 is connected differentially across these two voltages. Therefore the voltage V_2 will be equal in magnitude with V_1 but 180◦ out of phase and cancel out the dc biased voltage and ac sinusoidal voltage appear across the load.

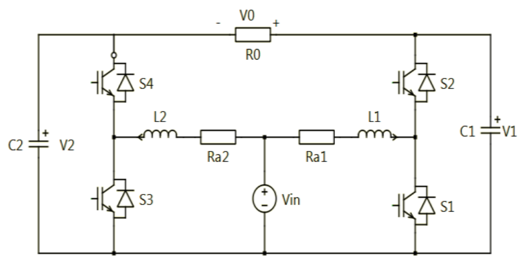


Figure 2. Dc–ac boost inverter

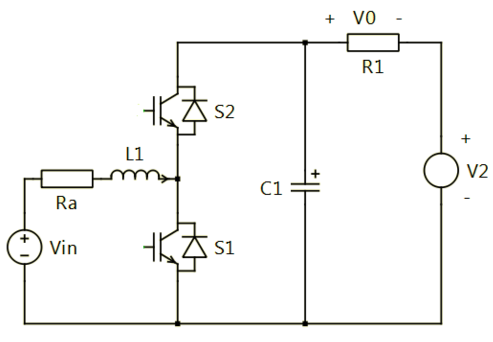


Figure 3. Equivalent circuit for the boost inverter

For the sake simplicity and to reduce complexity of calculation, the boost inverter is better understood through a single current bidirectional boost dc–dc converter with output V_1 and the equivalent circuit is shown in Figure 3. The state-space modeling of the equivalent circuit with state variables i_L and V_1 is given by,

$$\begin{bmatrix} d\dfrac{i_{L1}}{dt} \\ d\dfrac{V_1}{dt} \end{bmatrix} = \begin{bmatrix} \dfrac{-R_a}{L_1} & -\dfrac{1}{L_1} \\ \dfrac{1}{C_1} & -\dfrac{1}{C_1R_1} \end{bmatrix} \begin{bmatrix} i_{L1} \\ V_1 \end{bmatrix} + \begin{bmatrix} \dfrac{V_{in}}{L_1} \\ \dfrac{V_2}{C_1R_1} \end{bmatrix} + \begin{bmatrix} \dfrac{V_1}{L_1} \\ \dfrac{-i_{L1}}{C_1} \end{bmatrix} d \ . \quad (4)$$

To derive the small signal model of the boost inverter, the state variables are perturbed, $i_{L1} = I_{L1} + \hat{i}_{L1}$, $V_{in} = V_{in} + \hat{V}_{in}$, $V_1 = V_1 + \hat{V}_1$, $V_2 = V_2 + \hat{V}_2$ and $d = D + d$. Therefore Eqn (4) becomes,

$$d\frac{(I_{L1}+\hat{i}_{L1})}{dt} = \frac{-R_a}{L_1}(I_{L1}+\hat{i}_{L1}) - \frac{1}{L_1}(V_1+\hat{V}_1) + \frac{1}{L_1}(V_{in}+\hat{V}_{in}) + \frac{1}{L_1}(V_1+\hat{V}_1)(D+\hat{d}) \quad (5)$$

and

$$d\frac{(V_1+\hat{V}_1)}{dt} = \frac{1}{C_1}(I_{L1}+\hat{i}_{L1}) - \frac{1}{C_1R_1}(V_1+\hat{V}_1) + \frac{1}{C_1R_1}(V_2+\hat{V}_2) + \frac{1}{C_1}(I_{L1}+\hat{i}_{L1})(D+\hat{d}) \quad (6)$$

Now 'ac' and 'dc' quantities in Eqn (5) and (6) are equated and proceed with 'ac' equation neglecting second order 'ac' quantities, Eqn (5) & (6) becomes,

$$d\frac{\hat{i}_{L1}}{dt} = \frac{-R_a}{L_1}\hat{i}_{L1} - \frac{1}{L_1}\hat{V}_1 + \frac{1}{L_1}\hat{V}_{in} + \frac{1}{L_1}\hat{V}_1 D + \frac{1}{L_1}V_1\hat{d} \quad (7)$$

$$d\frac{\hat{V}_1}{dt} = \frac{1}{C_1}\hat{i}_{L1} - \frac{1}{C_1R_1}\hat{V}_1 + \frac{1}{C_1R_1}\hat{V}_2 + \frac{1}{C_1}\hat{i}_{L1}D + \frac{1}{C_1}I_{L1}\hat{d} \quad (8)$$

Taking Laplace Transformation, Eqn (7) & (8) becomes,

$$s.\hat{i}_{L1}(s) = \frac{-R_a}{L_1}\hat{i}_{L1}(s) - \frac{1}{L_1}\hat{V}_1(s) + \frac{1}{L_1}\hat{V}_{in}(s) + \frac{1}{L_1}\hat{V}_1(s)D + \frac{1}{L_1}V_1\hat{d}(s) \quad (9)$$

$$s.\hat{V}_1(s) = \frac{1}{C_1}\hat{i}_{L1}(s) - \frac{1}{C_1R_1}\hat{V}_1(s) + \frac{1}{C_1R_1}\hat{V}_2(s) + \frac{1}{C_1}\hat{i}_{L1}(s)D + \frac{1}{C_1}I_{L1}\hat{d}(s) \quad (10)$$

Rearranging in symmetrical sequence, Eqn (9) & (10) becomes,

$$(sL_1 + R_a)\hat{i}_{L1}(s) + (1-D)\hat{V}_1(s) = \hat{V}_{in}(s) + V_1\hat{d}(s) \quad (11)$$

$$(1-D)\hat{i}_{L1}(s) - (sC + 1/R_1)\hat{V}_1(s) = I_{L1}\hat{d}(s) - \hat{V}_2(s)/R_1 \quad (12)$$

From Eqn (11) & (12), the small signal model can be written as,

$$\begin{bmatrix} sL_1 + R_a & 1-D \\ 1-D & -(sC+1/R_1) \end{bmatrix} \begin{bmatrix} i_{L1}(s) \\ V_1(s) \end{bmatrix} + \begin{bmatrix} V_1 \\ I_{L1} \end{bmatrix} d(s) + \begin{bmatrix} 1 & 0 \\ 0 & -1/R \end{bmatrix} \begin{bmatrix} V_{in}(s) \\ V_2(s) \end{bmatrix} \quad (13)$$

From Eqn (13), the relation between output voltage, V_1 and duty cycle, d is given by,

$$\frac{V_1(s)}{d(s)} = \frac{-(L.I_{L1})s + R_a.I_{L1} + (1-D)V_1}{(L.C)s^2 + (\frac{L}{R} + C.R_a)s + \frac{R_a}{R} + (1-D)^2} \quad (14)$$

Therefore Eqn (14) is used for the design of controller for the boost inverter.

3.1 PID Controller

The PID controller is used to improve the dynamic response as well as to reduce or eliminate the steady-state error. The PID controller transfer function is given by,

$$C(s) = k_p + \frac{k_i}{s} + k_d s \quad (15)$$

The parameters used for the development of controller for boost inverter is tabulated in Table 2.

Table 2. Parameters of the boost inverter

Parameters	Range
Input voltage ,V_{in}	12~48 V
Rated output voltage, V_0	240 V
Rated output power, P_0	480 W
Switching frequency, f_s	20 kHz
Duty cycle, D	0.8
Inductance , L_1,L_2	96 μH
Capacitance , C_1,C_2	150 μF
Load, R_1	120Ω
Inductive resistance, Ra_1,Ra_2	0.001Ω

Using Table 2, the value of the output voltage, V_1 to the duty cycle, d is given by,

$$\frac{V_1(s)}{d(s)} = \frac{-0.00192s + 24.02}{1.44e^{-8}s^2 + 65.9722s + 6.9514e^5} \quad (16)$$

The boost inverter with PID controller is as shown in Figure 4.

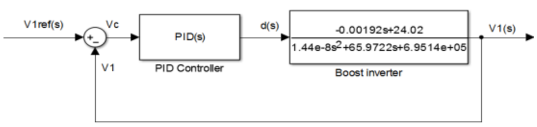


Figure 4. Boost inverter with PID controller

3.2 Estimation of PID Controller Performance

The performance criterion in the time domain is used for evaluating the PID controller. A set of good control parameters, k_p, k_i and k_d can yield a good step response that will result in minimization of performance criteria in the time domain. The performance criteria in the time domain includes the overshoot M_p, steady-state error E_{ss}, rise time t_r and settling time t_s. Therefore, performance criterion W(K) [16]is defined as,

$$W(K) = (1-e^{-\beta}).(M_p + E_{ss}) + e^{-\beta}.(t_s - t_r) \,. \qquad (17)$$

Therefore Eqn (16) is used to tune the PID controller.

4. PID Tuning using Optimization Techniques

The traditional PID tuning method undergoes difficulties when tuning a complex system [21]. To overcome these difficulties, EAs such as GA, ABC, ACO and PSO based PID tuning methods are employed and they are explained in the subsequent section.

4.1. GA-PID Controller

The genetic algorithm is developed based on the principle of survival of the fittest. The optimization process starts with random generation of population. Each individual in the population is called chromosomes and these represent a possible solution. The fundamental operators of GAs are selection, crossover and mutation [17]. The selection operator selects the best individuals in population. In crossover, the selected chromosomes exchange partially their information of the genes. The mutation operator creates a new individual by randomly mutating a randomly selected part of a selected chromosome.

The steps of implementing GA for PID controller are as follows:

Step 1: Generate random population of individuals, k called chromosomes of fixed size, n where each individual contain the controller optimal parameters k_p, k_i, k_d representing a possible solution. The parameters of PID at the initial stage could make the system unstable. Therefore, the range of the controller parameters is selected such that the system remains stable within this range.

Step 2: Evaluate the fitness function of each chromosome using Eqn (17).

Step 3: The fittest members is selected for the reproduction.

Step 4: Probabilistic method is used for reproduction.

Step 5: Crossover operation is performed on the chromosomes to obtain the offspring.

Step 6: Mutation operation is operated on the chromosomes.

Step 7: Repeat from step 2 until a predefined convergence criterion is met.

4.2. ABC-PID Controller

The ABC algorithm is inspired from the intelligent foraging behavior of the honeybees [18]. In order to find the best solution, the algorithm defines three classes of bees: employer bees, onlooker bees, and scout bees. The employed bee searches for the available food sources and gathers the required information. The onlooker bee makes a decision to choose the good food sources by sharing the information of employed bee and they further search for foods.

If a solution representing a food source cannot be improved by a predetermined number of trials, associated food source has been exhausted by the bees and then the employed bee of this food source becomes a scout bee. The position of the abandoned food source (solution) is then replaced with a randomly produced solution. The employed bees search for new neighbour food source near their hive. A new position $k_{i,j}$ of the solution is generated using,

$$k_{i,j}^{new} = k_{i,j}^{old} + \phi_{i,j}(k_{i,j}^{old} - k_{l,j}) \qquad (18)$$

where, l is a randomly chosen index from the population {1,2,3,…,SN} different than i,j is a randomly chosen index from {1,2,3,….D} , SN is the number of food sources, D is the problem dimension, and $\varphi_{i,j}$ is a uniformly distributed random number within [-1, 1]. ABC uses a greedy selection operator. An onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, given by,

$$p_i = \frac{fit_i}{\sum_{m=1}^{SN} fit_m} \qquad (19)$$

where, fit_i is the fitness value of the ith solution, which is proportional to the amount of nectar in the food source in the ith position. When a food source (solution) cannot be improved anymore, then the scout bee helps the colony to randomly generate new solutions.

The steps of implementing ABC for PID controller are as follows:

Step 1: Initialize the food-source positions k with size n (solutions population), where n=1, 2,..., E_b.

Step 2: Calculate the amount of nectar in the population by means of their fitness values using,

$$fitness_i = \frac{1}{1+obj_i} \qquad (20)$$

where, obj_i represents the response of (17) at solution.

Step 3: Produce neighbor solutions for the employed bees by using Eqn (18) and evaluate them as indicated by step 2.
Step 4: Apply the selection process.
Step 5: If all onlooker bees are distributed, go to stop. Otherwise, go to step 6.
Step 6: Calculate the probability values p_i for the solutions x_i using Eqn (19).
Step 7: Produce neighbor solutions for the selected onlooker bee, depending on the value, using Eqn (18) and evaluate from step 2.
Step 8: Determine the abandoned solution for the scout bees, if it exists replace it with a completely new solution using,

$$k_i^{j(new)} = \min(k_i^j) + \phi_{i,j}[\max(k_i^j) - \min(k_i^j)] \tag{21}$$

and evaluate from step 2.

4.3 ACO-PID Controller

ACO was inspired by the behavior of real ants. The medium that is used to communicate information among individual ants regarding paths is pheromone. A moving ant lays some pheromone on the ground, thus marking the path. The pheromone, while gradually dissipating over time, is reinforced as other ants use the same trail. Therefore, efficient trails increase their pheromone level over time while poor ones reduce to nil. In particular, an ant constructs a candidate solution to a problem by iteratively adding solution components to its partial solution in a stochastic fashion [19]. The probability that an ant k chooses to visit node j after node i is given by,

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^m} [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta} \tag{21}$$

where, τ_{ij} is the pheromone associated with adding the edge (i, j) to the current partial tour, η_{ij} is a static greedy measure of the "goodness" of edge (i, j) called heuristic information, and N_i^m denotes the set of feasible choices available for ant k located in node i given its current partial solution. After a number of ants have constructed a solution each, one or more of these solutions are used to perform other actions, such as local search, to further improve solutions before updating the pheromone information in such a way so as to bias future choices toward high quality solutions.

The steps of implementing ACO for PID controller are as follows:
Step 1: Initialize randomly potential solutions of the parameters k_p, k_i and k_d using uniform distribution. Initialize the pheromone trail and the heuristic value.
Step 2: Place the A^{th} ant on the node. Compute the heuristic value associated on the objective function.
Step 3: Use pheromone evaporation to avoid unlimited increase of pheromone trails and allow the forgetfulness of bad choices given by,

$$\tau_{ij} = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta\tau_{ij}^A(t) \tag{22}$$

where, $\Delta\tau_{ij}^A$ the quantity of pheromone on each path, NA: number of ants, ρ: the evaporation rate $0 < \rho \leq 1$.
The quantity of pheromone $\Delta\tau_{ij}^A$ on each path may be defined as follows:

$$\Delta\tau_{ij}^A = \begin{cases} \dfrac{L^{\min}}{L^A} & if\ i, j \in T^A \\ 0 & \end{cases} \tag{23}$$

where, L_A = the value of the objective function found by the ant A. L_{min} = the best solution carried out by the set of the ants until the current iteration.
Step 4: Evaluate the fitness function using Eqn (17).
Step 5: calculate the optimum values of the optimization parameters.
Step 6: Update the pheromone, according to the optimum solutions calculated at step 5: Iterate from step 2 until the maximum of iterations is reached.

4.4 PSO-PID Controller

The particle swarm optimization (PSO) is basically developed from research on social behavior of bird flocking and fish schooling. The PSO algorithm maintains a swarm of individuals (called particles), where each particle represents a candidate solution. Particles follow a simple behavior: velocity is dynamically adjusted according to particles own experience and its companion's experience [10]. The coordinates of the particles associated with the best solution is called P_{best} and the overall best value coordinates of the particles associated with the global solution is called G_{best}. Particle position, k is adjusted using

$$k_i^{t+1} = k_i^t + v_i^{t+1} \tag{24}$$

where is the velocity and is calculated using,

$$v_i^{t+1} = w.v_i^t + c_1 r_1 (P_{best} - k_i^t) + (G_{best} - k_i^t) \tag{25}$$

where, w is the inertia weight, c_1 and c_2 are the acceleration coefficients, r_1, $r_2 \in U(0,1)$ are random numbers, P_{best} is the personal best position of particle i, and G_{best} is the best position of the particles.

The steps of implementing PSO for PID controller are as follows:
Step 1: Initialize the particle k with number of particles, n. Each particle, k contain the controller optimal parameters k_p, k_i, k_d. Therefore its dimension is chosen as n×3.
Step 2: For each individual k of the n particle, calculate the fitness function.
Step 3: calculate the P_{best} using the fitness function and the global best value of P_{best} as G_{best}.
Step 4: Update the velocity of the individual particle k according to Eqn (25)
Step 5: Update the position using Eqn (24)
Step 6: if the convergence is obtained then goto step 7, else goto step 2.
Step 7: the G_{best} of the convergence iteration contains the optimal value controllers' parameters.

5. Simulation Results

To verify the efficiency of the PID controller, a boost inverter PV system is tested. The lower and upper bounds of the controller parameters are shown in Table 3. Figure 5 shows the step response of the boost inverter without a PID controller. Figure 6 shows the bode plot of the boost inverter without a PID controller. The performance criteria in the time domain are M_p=88.3%, E_{ss}=2.4e^3, t_s=0.117 s, t_r=0.00129 s and β = 0.8.

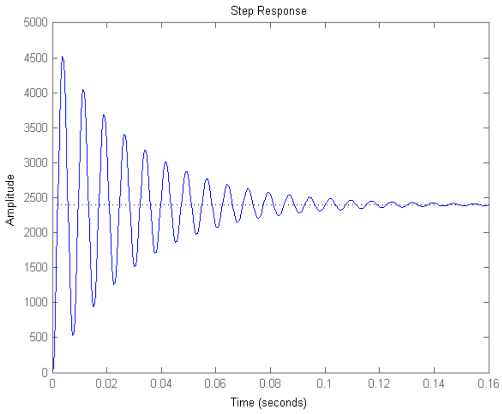


Figure 5. Step response of a boost inverter without PID controller

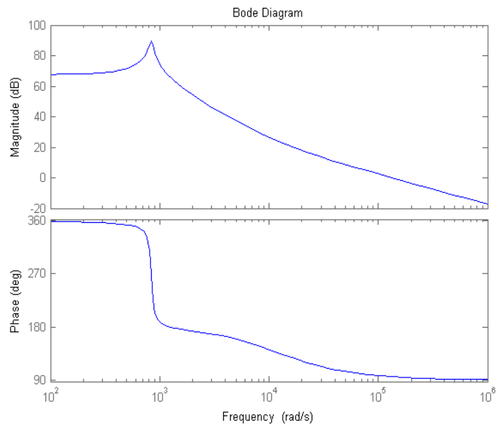


Figure 6. Bode plot of a boost inverter without PID controller

Table 3. Range of controller parameters

Controllers parameters	Minimum value	Maximum value
K_p	0	1
K_i	0	0.1
K_d	0	1

5.1. Performance of the PID Controller

Figure 7 shows the step response of the boost inverter with GA, ACO, ABC, and PSO based PID controller.

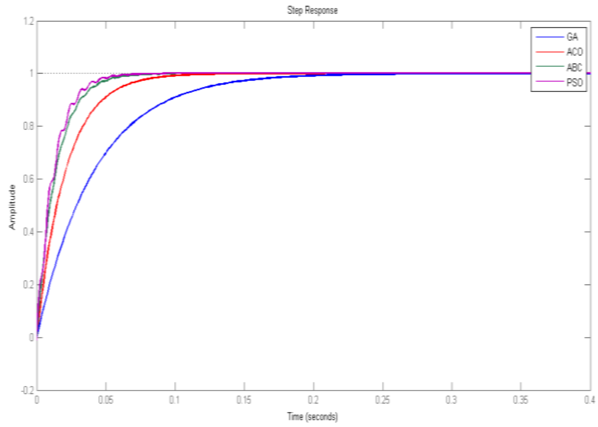
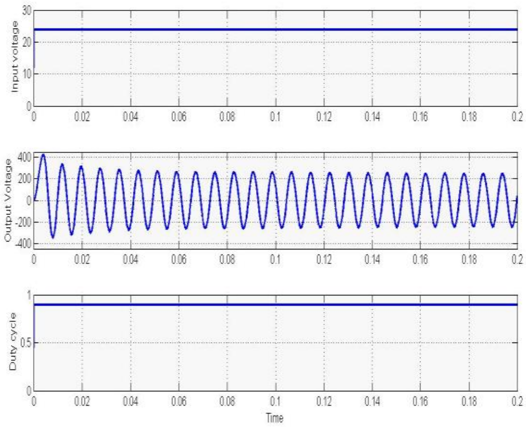
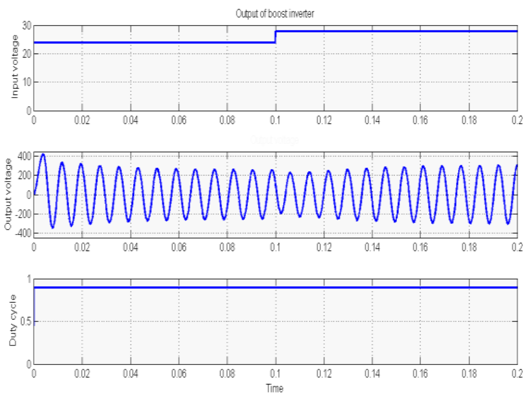


Figure 7. Step response of a boost inverter with PID controller

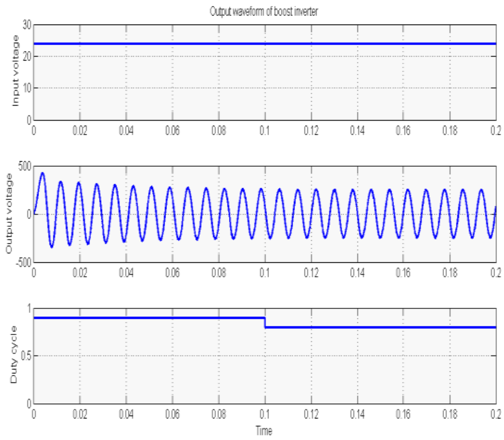
The simulation results that showed the best solution are summarized in Table 4.



(a) Output voltage response for step change in input voltage from 0V to 12V



(b) Output voltage response for step change in input voltage from24V to 28V



(c) Output voltage for change in reference voltage from 24 V to 18 V

Figure 8. Measure results for closed loop output voltage control

From Figure 7, the PSO based PID controller prompt convergence and obtain good evaluation value among the other EA based PID controller like GA, ACO and ABC. From the Table 4, the PSO gives better convergence and obtain good evaluation value when compared with Bees-GA PID and PSO-PID based controller for boost inverter. The output voltage response for step change in input voltage, reference voltage for the PSO-PID controller is shown in Figure 8. It shows a good response for a step change in voltage. This result shows that the PSO based controller can search optimal PID controller parameters quickly and efficiently. Four controllers and their performance evaluation criteria in the time domain were implemented by Matlab and control system toolbox, and executed on an i7 processor personal computer with 4-GB RAM.

Table 4. Types of controller and its parameters

Type of controller	K_p	K_i	K_d	M_p (%)	E_{ss}	t_s	t_r
GA-PID	9.5e-7	0.01001	1.44e-8	0	0	0.0913	0.0811
ACO-PID	4.5e-7	0.02	3e-8	0	0	0.163	0.0454
ABC-PID	3e-7	0.03001	5e-8	0	0	0.08	0.0298
PSO-PID	1e-7	0.03401	8e-8	0	0	0.08	0.0281
Bees-GA PID [20]	0.326	8.87	0.0012	0	0.13	0.16	0.0356
PSO-PID [21]	114.32	11.56	0	0.03	0.03	0.00899	0.16

6. Conclusion

This paper presents a design method for determining the PID controller parameters using the evolutionary algorithms. The proposed method integrates the evolutionary algorithms with the time-domain performance criterion into evolutionary algorithms based PID controller. Through the simulation of a boost inverter system, the results show that the proposed controller can perform an efficient search for the optimal PID controller parameters. In addition, comparison is made between GA, ABC, ACO and PSO, in order to verify its superiority. It is clear from the comparison results that the PSO algorithm can obtain higher quality solution with better computation efficiency. Therefore, the PSO algorithm has more robust stability and efficiency, and can solve the searching and tuning problems of PID controller parameters more easily and quickly.

7. References

[1] W. Rong-Jong, W. Wen-Hung, and L. Chung-You, "High-Performance Stand-Alone Photovoltaic Generation System," IEEE Transactions on Industrial Electronics, vol. 55, pp. 240-250, 2008.

[2] Z. Zheng, X. Ming, C. Qiaoliang, L. Jih-Sheng, and C. Younghoon, "Derivation, Analysis, and Implementation of a Boost -Buck Converter-Based High-Efficiency PV Inverter," IEEE Transactions on Power Electronics, vol. 27, pp. 1304-1313, 2012.

[3] H. S. H. Chung, K. K. Tse, S. Y. R. Hui, C. M. Mok, and M. T. Ho, "A novel maximum power point tracking technique for solar panels using a SEPIC or Cuk converter," IEEE Transactions on Power Electronics, vol. 18, pp. 717-724, 2003.

[4] M. A. Al-Saffar, E. H. Ismail, A. J. Sabzali, and A. A. Fardoun, "An Improved Topology of SEPIC Converter With Reduced Output Voltage Ripple," IEEE Transactions on Power Electronics, vol. 23, pp. 2377-2386, 2008.

[5] J. E. Baggio, H. L. Hey, H. A. Grundling, H. Pinheiro, and J. R. Pinheiro, "Isolated interleaved-phase-shift-PWM DC-DC ZVS converter," IEEE Transactions on Industry Applications, vol. 39, pp. 1795-1802, 2003.

[6] Y. Zhongming, P. K. Jain, and P. C. Sen, "A Two-Stage Resonant Inverter with Control of the Phase Angle and Magnitude of the Output Voltage," IEEE Transactions on Industrial Electronics, vol. 54, pp. 2797-2812, 2007.

[7] Z. Xiaotian and J. W. Spencer, "Study of Multisampled Multilevel Inverters to Improve Control Performance," IEEE Transactions on Power Electronics, vol. 27, pp. 4409-4416, 2012.

[8] N. Nho-Van, N. Bac-Xuan, and L. Hong-Hee, "An Optimized Discontinuous PWM Method to Minimize Switching Loss for Multilevel Inverters," IEEE Transactions on Industrial Electronics, vol. 58, pp. 3958-3966, 2011.

[9] S. Jain and V. Agarwal, "A Single-Stage Grid Connected Inverter Topology for Solar PV Systems with Maximum Power Point Tracking," IEEE Transactions on Power Electronics, vol. 22, pp. 1928-1940, 2007.

[10] W. Chien-Ming and C. Teng-Jen, "Novel single-stage half-bridge series-resonant buck-boost inverter," IEEE

Transactions on Aerospace and Electronic Systems, vol. 40, pp. 1262-1270, 2004.
[11] R. A. Mastromauro, M. Liserre, and A. Dell'Aquila, "Control Issues in Single-Stage Photovoltaic Systems: MPPT, Current and Voltage Control," IEEE Transactions on Industrial Informatics, vol. 8, pp. 241-254, 2012.
[12] R. O. Caceres and I. Barbi, "A boost DC-AC converter: analysis, design, and experimentation," IEEE Transactions on Power Electronics, vol. 14, pp. 134-141, 1999.
[13] D. Cortes, N. Vazquez, and J. Alvarez-Gallegos, "Dynamical Sliding-Mode Control of the Boost Inverter," IEEE Transactions on Industrial Electronics, vol. 56, pp. 3467-3476, 2009.
[14] P. Sanchis, A. Ursaea, E. Gubia, and L. Marroyo, "Boost DC-AC inverter: a new control strategy," IEEE Transactions on Power Electronics, vol. 20, pp. 343-353, 2005.
[15] Z. Wei, D. D. C. Lu, and V. G. Agelidis, "Current Control of Grid-Connected Boost Inverter with Zero Steady-State Error," IEEE Transactions on Power Electronics, vol. 26, pp. 2825-2834, 2011.
[16] G. Zwe-Lee, "A particle swarm optimization approach for optimum design of PID controller in AVR system," IEEE Transactions on Energy Conversion, vol. 19, pp. 384-391, 2004.
[17] M. J. Neath, A. K. Swain, U. K. Madawala, and D. J. Thrimawithana, "An Optimal PID Controller for a Bidirectional Inductive Power Transfer System Using Multiobjective Genetic Algorithm," IEEE Transactions on Power Electronics, vol. 29, pp. 1523-1531, 2014.
[18] F. S. Abu-Mouti and M. E. El-Hawary, "Optimal Distributed Generation Allocation and Sizing in Distribution Systems via Artificial Bee Colony Algorithm," IEEE Transactions on Power Delivery, vol. 26, pp. 2090-2101, 2011.
[19] M. Lopez-Ibanez and T. Stutzle, "The Automatic Design of Multiobjective Ant Colony Optimization Algorithms," IEEE Transactions on Evolutionary Computation, vol. 16, pp. 861-875, 2012.
[20] K. Sundareswaran and V. T. Sreedevi, "Boost Converter Controller Design Using Queen-Bee-Assisted GA," IEEE Transactions on Industrial Electronics, vol. 56, pp. 778-783, 2009.
[21] K. Y. Abhimanyu, V. Mehra, A. Ray, A. Markana and M. Lokhande, "Paralleled DC Boost Converters with Feedback Control using PSO Optimization Technique for Photovoltaic Module Application," International Journal of Computer Applications, vol. 84, pp. 12-18, 2013.

B.Goldvin Sugirtha Dhas (M'08) was born in Neyyoor, India, in 1988. He received the B.E. degree in electrical and electronics engineering from the Anna University, Chennai, India, in 2010, and the M.E. degree in power electronics and drives from the Anna University, Chennai, India, in 2012.

He is currently working toward the Ph.D. degree in the Department of Electrical and Electronics Engineering, Anna University, Regional Centre, Coimbatore, India.

His current research interests include power electronics, renewable energy systems, electrical machines, electrical drives and soft computing techniques.

S.N.Deepa (M'13) was born in Coimbatore, India, in 1978. She received the B.E. degree in electrical and electronics engineering from the Anna University, Chennai, India, in 1999, and the M.E. degree in control systems from the Anna University, Chennai, India, in 2004, and the Ph.D. degree in electrical engineering from the Anna University, Chennai, India, in 2008.

She is currently working as Associate Professor and Head of the Department, Dept. of Electrical and Electronics Engineering, Anna University, Regional Centre: Coimbatore, Tamil Nadu, India.

Her Research areas include linear and non-linear control system design and analysis, Modelling and simulation, Soft Computing, Evolutionary Strategies and Adaptive Control Systems.